

[illegible]

```
SSSSSSSS MM MM BBBB BBBB RRRRRRRR EEEEEEEEE QQQQQQ
SSSSSSSS MM MM BBBB BBBB RRRRRRRR EEEEEEEEE QQQQQQ
SS SS MMM MMM BB BB RR RR EE EE QQ QQ
SS SS MMM MMM BB BB RR RR EE EE QQ QQ
SS SS MM MM BB BB RR RR EE EE QQ QQ
SSSSSS MM MM BBBB BBBB RRRRRRRR EEEEEEEEE QQQQQQ
SSSSSS MM MM BBBB BBBB RRRRRRRR EEEEEEEEE QQQQQQ
SS MM MM BB BB RR RR EE EE QQ QQ
SS MM MM BB BB RR RR EE EE QQ QQ
SS MM MM BB BB RR RR EE EE QQ QQ
SSSSSS MM MM BBBB BBBB RRRRRRRR EEEEEEEEE QQQQQQ
SSSSSS MM MM BBBB BBBB RRRRRRRR EEEEEEEEE QQQQQQ
```

```
RRRRRRRR EEEEEEEEE QQQQQQ
RRRRRRRR EEEEEEEEE QQQQQQ
RR RR EE QQ QQ
RR RR EE QQ QQ
RR RR EE QQ QQ
RRRRRRRR EEEEEEEEE QQQQQQ
RRRRRRRR EEEEEEEEE QQQQQQ
RR RR EE QQ QQ
RR RR EE QQ QQ
RR RR EE QQ QQ
RR RR EE QQ QQ
RR RR EEEEEEEEE QQQQQQ
RR RR EEEEEEEEE QQQQQQ
```

Require file for print symbiont facility

Version: 'V04-000'

COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
ALL RIGHTS RESERVED.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

FACILITY:

Symbiont Services.

ABSTRACT:

Macro and literal definitions for symbionts.

AUTHOR: G. Robert, CREATION DATE: 01-May-1983

MODIFIED BY:

3B-007	RRB3007	Rowland R. Bradley	20-Jul-1984
	Change PSMS_ messages NOMOREITEMS, INVSTRLEV, and INVSIMNBR to SMB\$ NOMOREITEMS, SMB\$ INVSTRLEV, and SMB\$ INSTRMNR, respectively. Delete messages PSMS_INVSCB, PSMS_REQNOTSUP, PSMS_INVREQCOD.		
3B-006	GRR3006	Gregory R. Robert	29-Apr-1984
	Added PSMS_FLUSH		
3B-005	RRB0014	Rowland R. Bradley	27-Apr-1984
	Remove the Task_Flag macro.		
3B-004	RRB0013	Rowland R. Bradley	21-Feb-1984
	Add new STRUCTURE definition for PAGE.		


```
3B-003 GRR3004 Gregory R. Robert 23-Aug-1983
Bugfixes, page_setup_modules, form_setup_modules,
sheet_feed, symbiont_initiated_pause_task and stop_stream,
hangup code, read and write item services

3B-003 GRR3003 Gregory R. Robert 03-Aug-1983
Fixes for new design.

3B-002 GRR3002 Gregory R. Robert 29-Jul-1983
Added several macros to access symbiont tables.

3B-001 GRR3001 Gregory R. Robert 23-Jun-1983
Fixed offset_table macro, added item_present macro,
fixed some bugs, added several literals.

3B-000 GRR3000 Gregory R. Robert 27-May-1983
Original version.
```

```
! Define program section standard names and attributes
```

```
PSECT
```

```
CODE = CODE,
PLIT = CODE,
OWN = DATA,
GLOBAL = DATA
;
```

```
! Check that structure id's have a common byteoffset
```

```
$ASSUME ($BYTEOFFSET (IOB_L_STRUCTURE), EQL, $BYTEOFFSET (PSM$L_STRUCTURE))
```

```
! Check that a quadword of item code flags is adequate
```

```
$ASSUME (SMBMSGK_MAX_ITEM_CODE - 1, LSS, 64)
```

```
! Delcare useful builtin Bliss functions
```

```
BUILTIN
```

```
CALLG,
FFS,
INSQUE,
LOCC,
MOVCS,
MOVCS,
MOVTUC,
REMOUE,
TESTBITCC,
TESTBITCS,
TESTBITSC,
TESTBITSS
;
```

```
! Declare special linkages
```

```
! LINKAGE
  ANALYZE_LINKAGE = JSB (REGISTER=0; REGISTER=0, REGISTER=1),
```

```
! CAN'T USE THIS UNTIL UNDERSTAND HOW TO PRESERVE R2-R4 INCLUSIVE
! FREE_LINKAGE = JSB (REGISTER=0)
```

```
! Declare common external routines
```

```
EXTERNAL ROUTINE
  BASSEDIT,
```

```
  LBR$CLOSE,
  LBR$GET_RECORD,
  LBR$INI_CONTROL,
  LBR$LOOKUP_KEY,
  LBR$OPEN,
  LBR$RET_RMSSTV,
  LBR$SET_LOCATE,
```

```
  LIB$TRIM_FILESPEC,
  LIB$GET_VM,
  LIB$FREE_VM,
```

```
  STR$ANALYZE_SDESC,
  STR$ANALYZE_SDESC_R1 : ANALYZE_LINKAGE,
  STR$APPEND,
  STR$CONCAT,
  STR$COPY_DX,
  STR$COPY_R,
  STR$FREE_DX,
  STR$FREE1_DX_R4 : FREE_LINKAGE,
  STR$GET1_DX,
  STR$LEFT,
  STR$PREFIX,
  STR$RIGHT
;
```

```
EXTERNAL LITERAL
  PSMS$HANGUP_DISPATCH_ENTRY
;
```

```
! Private messages
```

```
EXTERNAL LITERAL
  PSMS_BUFFEROVF,
  PSMS_EOF,
  PSMS_ESCAPE,
  PSMS_FLUSH,
  PSMS_FUNNOTSUP,
  PSMS_INVITMCO,
  PSMS_INVVMSOSC,
```

```

PSMS_MODNOTFND,
PSMS_NEWPAGE,
PSMS_NOFILEID,
PSMS_OSCTOOLON,
PSMS_PENDING,
PSMS_SUSPEND,
PSMS_TOOMANYLEV,
SMB$_INVSTMNBR,
SMB$_INVSTRLEV,
SMB$_NOMOREITEMS
;

```

```

! Shared messages

```

```

$SHR_MSGDEF (PSM, PSMSK FACILITY, LOCAL,
(BADLOG(PC, SEVERE), - logic error with PC value
(CLOSEIN, ERROR), - unable to close input
(OPENIN, ERROR), - unable to open or connect to input
(READERR, ERROR), - error reading
(WRITEERR, ERROR) - error writing
);

```

```

! Define structures useful for accessing parameters passed by reference

```

```

STRUCTURE
$BYTE [] = $BYTE <0,08,0>,
$SIGNED_BYTE [] = $SIGNED_BYTE <0,08,1>,

$WORD [] = $WORD <0,16,0>,
$SIGNED_WORD [] = $SIGNED_WORD <0,16,1>,

$LONGWORD [] = $LONGWORD <0,32,0>,
$SIGNED_LONGWORD [] = $SIGNED_LONGWORD <0,32,1>
;

```

```

! Define structures useful for referencing the "page" of information

```

```

STRUCTURE
PAGE_ARRAY[I, J, K; N, M, UNITS=1] = ! default is byte referencing
[M * N * UNITS]
(PAGE_ARRAY + (J * K + I) * UNITS)<0,8,0>
;

```

```

! Message Item Table (MIT) and Service Routine (SRV) table building macros

```

```

MACRO

```

```

MIT_PRESET [TAG, RESET, TYPE, ITEM] =
[NAME ('SMBMSG$K', ITEM), MIT_B_TYPE] = NAME ('MIT_K', TYPE),
[NAME ('SMBMSG$K', ITEM), MIT_V_RESET] = %IF %NULL (RESET) %THEN 1 %ELSE RESET %FI,
[NAME ('SMBMSG$K', ITEM), MIT_W_OFFSET] =
$BYTEOFFSET (NAME ('PSM$', TAG, '-', ITEM))
% ;

```

```

MACRO

```



```

SRV_PRESET [SERVICE, USER, TYPE] =
  [XNAME ('PSMSK', SERVICE), SRV_A_SERVICE] =
    XNAME ('PSMSK', SERVICE),
  [XNAME ('PSMSK', SERVICE), SRV_V_USER_ALLOWED] =
    %IF %NULL (USER) %THEN NO_USER
    %ELSE USER %FI,
  [XNAME ('PSMSK', SERVICE), SRV_B_SERVICE_TYPE] =
    %IF %NULL (TYPE) %THEN SRV_R_GENERAL_SERVICE
    %ELSE XNAME ('SRV_K_', TYPE, '_SERVICE') %FI
% ;

```

! General purpose macros

MACRO

```

ACC_DATA (ITEM) =
  %BBLCK [SCB[PSMST_ACCOUNTING_AREA], XNAME ('SMBMSG$L_', ITEM)]
%,

BLINK (QUEUE_HEADER) =
  VECTOR [QUEUE_HEADER, 1]
%,

CLEAR_QUAD_ (QUAD) =
  BEGIN
    VECTOR [QUAD, 0] = 0;
    VECTOR [QUAD, 1] = 0;
  END
%,

CLEAR_STRING_ (DESC) =
  BEGIN
    IF .DESC_CLASS_ (DESC) LEQU DSC$K_CLASS_S ! 0 or 1
    THEN
      INIT_DYN_DESC_ (DESC)
    ELSE
      IF .DESC_SIZE_ (DESC) NEQ 0
      THEN
        STR$FREE1_DX (DESC)
      END
    END
%,

CODEERR =
  SIGNAL_STOP (PSM$BADLOGICPC, 1) ! PC implied 3rd arg
%,

COPY_DX_DX_ (FROM_DESC, TO_DESC) =
  SIGNAL_IF_ERROR_ (STR$COPY_DX (TO_DESC, FROM_DESC))
%,

COPY_R_DX_ (FROM_SIZE, FROM_ADDRESS, TO_DESC) =
  SIGNAL_IF_ERROR_ (STR$COPY_R (TO_DESC, FROM_SIZE, FROM_ADDRESS))
%,

COPY_QUAD_ (FROM_QUAD, TO_QUAD) =
  BEGIN

```

```
    VECTOR [TO_QUAD, 0] = .VECTOR [FROM_QUAD, 0];
    VECTOR [TO_QUAD, 1] = .VECTOR [FROM_QUAD, 1];
END
%,
DECREMENT_ (VALUE) =
    BEGIN
        VALUE = .VALUE - 1;
    END
%,
DESC_ADDR_ (DESC) =
    $BLOCK [DESC, DSC$A_POINTER]
%,
DESC_CLASS_ (DESC) =
    $BLOCK [DESC, DSC$B_CLASS]
%,
DESC_SIZE_ (DESC) =
    $BLOCK [DESC, DSC$W_LENGTH]
%,
$DYNAMIC DESC =
    $BLOCK [8] PRESET (
        [DSC$W_LENGTH]      = 0,
        [DSC$B_DTYPE]       = DSC$K_DTYPE_T,
        [DSC$B_CLASS]       = DSC$K_CLASS_D,
        [DSC$A_POINTER]     = 0
    )
%,
FLINK_ (QUEUE_HEADER) =
    VECTOR [QUEUE_HEADER, 0]
%,
INCREMENT_ (VALUE) =
    BEGIN
        VALUE = .VALUE + 1
    END
%,
INIT_DYN_DESC_ (DESC) =
    BEGIN
        BIND X_DESC = DESC: $BLOCK;
        X_DESC [DSC$W_LENGTH] = 0;
        X_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
        X_DESC [DSC$B_CLASS] = DSC$K_CLASS_D;
        X_DESC [DSC$A_POINTER] = 0;
    END
%,
INIT_QUEUE_HEADER_ (QUEUE_HEADER) =
    BEGIN
        FLINK_ (QUEUE_HEADER) = QUEUE_HEADER;
        BLINK_ (QUEUE_HEADER) = QUEUE_HEADER;
```



```

X, END
INIT_STAT_DESC_ (DESC, LENGTH, POINTER) =
  BEGIN
    BIND X_DESC = DESC: $BLOCK;
    X_DESC[DSC$W_LENGTH] = LENGTH;
    X_DESC[DSC$B_DTYPE] = 0;
    X_DESC[DSC$B_CLASS] = 0;
    X_DESC[DSC$A_POINTER] = POINTER;
  END
X,
INSERT_HEAD (ENTRY_ADDR_, QUEUE_HEADER) =
  INSQUE (ENTRY_ADDR_, .FLINK_(QUEUE_HEADER))
X,
INSERT_TAIL (ENTRY_ADDR_, QUEUE_HEADER) =
  INSQUE (ENTRY_ADDR_, .BLINK_(QUEUE_HEADER))
X,
ITEM_PRESENT (ITEM_CODE) =
  BITVECTOR [SCB[PSM$Q_ITEM_FLAGS], %NAME ('SMBMSG$K_', ITEM_CODE)]
X,
OFFSET_TABLE_REPEAT_ [OFFSET, FIRST_BIT, SIZE, SIGN] =
  OFFSET
X,
PARAMETER_INDEX_ [] =
  BUILTIN NUL[PARAMETER;
  LITERAL PARAMETER_INDEX_REPEAT_ (%REMAINING)
X,
PARAMETER_INDEX_REPEAT [PARAMETER] =
  %NAME ('_P_', PARAMETER) = %COUNT + 1
X,
PARAMETER_PRESENT (PARAM) =
  NOT NULLPARAMETER (%NAME ('_P_', PARAM))
X,
PRINT_FLAG (FLAG_NAME) =
  $BLOCK [SCB[PSM$L_PRINT_FLAGS], %NAME ('SMBMSG$V_', FLAG_NAME)]
X,
READ_CHAR_ =
  BEGIN
    DECREMENT (SCB_SIZE_ (INPUT_RECORD));
    CH$RCHAR_A (SCB_ADDR_ (INPUT_RECORD))
  END
X,
REMOVE_HEAD (RESULT, QUEUE_HEADER) =
  REMQUE (.FLINK_(QUEUE_HEADER), RESULT)
X,

```

```
REMOVE_TAIL_ (RESULT, QUEUE_HEADER) =  
  REMOVE_T.BLINK_ (QUEUE_HEADER), RESULT)  
X,  
REQUEST_FLAG_ (FLAG_NAME) =  
  $BBLOCK [SCB[PSM$L_REQUEST_CONTROL], XNAME ('SMBMSG$V_', FLAG_NAME)]  
X,  
SEPARATE_FLAG_ (FLAG_NAME) =  
  $BBLOCK [SCB[PSM$L_SEPARATION_CONTROL], XNAME ('SMBMSG$V_', FLAG_NAME)]  
X,  
SERVICE_LIST_ (SERVICE) =  
  BITVECTOR [SCB[PSM$L_SERVICE_LIST], XNAME ('PSM$K_', SERVICE)]  
X,  
RETURN_IF_ERROR_ (ACTION) =  
  BEGIN  
    LOCAL STATUS;  
    STATUS = ACTION;  
    IF NOT .STATUS THEN RETURN (.STATUS);  
  .STATUS  
  END  
X,  
SET_DYN_DESC_ (DESC) =  
  BEGIN  
    $BBLOCK [SCB[XNAME ('PSM$Q_', DESC)], DSC$B_DTYPE] = DSC$K_DTYPE_T;  
    $BBLOCK [SCB[XNAME ('PSM$Q_', DESC)], DSC$B_CLASS] = DSC$K_CLASS_D;  
  END  
X,  
SIGNAL_IF_ERROR_ (ACTION) =  
  BEGIN  
    LOCAL STATUS;  
    STATUS = ACTION;  
    IF NOT .STATUS THEN SIGNAL (.STATUS);  
  .STATUS  
  END  
X,  
PSM$L_ = 0,0,32,0  
X,  
SCB_ADDR_ (DESC) =  
  DESC_ADDR_ (SCB [ XNAME ('PSM$Q_', DESC) ] )  
X,  
SCB_CLASS_ (DESC) =  
  DESC_CLASS_ (SCB [ XNAME ('PSM$Q_', DESC) ] )  
X,  
SCB_SIZE_ (DESC) =  
  DESC_SIZE_ (SCB [ XNAME ('PSM$Q_', DESC) ] )  
X,
```

```
WRITE_CHAR_ (CHAR) =  
  BEGIN  
    CH$WCHAR_A (CHAR, SCB_ADDR (OUTPUT_BUFFER));  
    DECREMENT_ (SCB_SIZE_ (OUTPUT_BUFFER));  
  END
```

```
%  
:
```


0309 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY